```
!pip install numpy pandas matplotlib
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('car_prices.csv')
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.26.4)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.8.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.2.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.
```

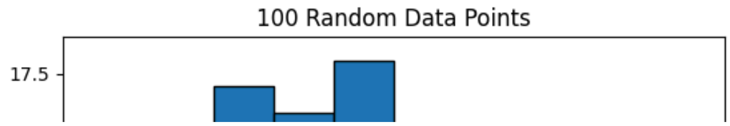
```
df['make'] = df['make'].str.lower()
df['model'] = df['model'].str.lower()
ford_f150 = df[(df['make'] == 'ford') & (df['model'] == 'f150')].copy()
ford_f150['sellingprice'] = pd.to_numeric(ford_f150['sellingprice'].astype(str).str.replace(',', ''), errors='coerce')
ford_f150 = ford_f150.dropna(subset=['sellingprice'])
ford_selling_prices = ford_f150['sellingprice']
avg_ford_f150_price = ford_selling_prices.mean()

print("Average selling price for all Ford F150s: ", avg_ford_f150_price)
```

```
Average selling price for all Ford F150s:  8222.177419354839
```

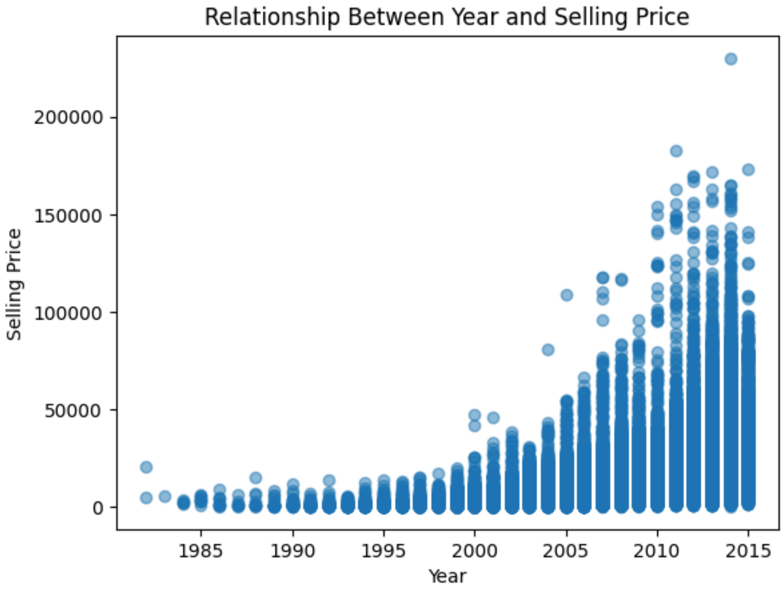
```
year_summary = df.groupby('year').size().reset_index(name = 'counts')
mcy_info = year_summary.sort_values(by='counts', ascending = False).iloc[0]
mcy = mcy_info['year']
mcy_data = df[df['year'] == mcy]
mc_make_model = mcy_data.groupby(['make', 'model']).size().reset_index(name = 'counts')
mc_make_model = mc_make_model.sort_values(by = 'counts', ascending = False).iloc[0]

print(f"The year with the most used car data: {mcy}")
print(f"The most common make and model for that year: {mc_make_model['make']} {mc_make_model['model']}")
```

```
The year with the most used car data: 2012
The most common make and model for that year: nissan altima
```

```
data_points = np.random.normal(loc=0, scale=1, size=100)

plt.hist(data_points, bins=10, edgecolor='black')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('100 Random Data Points')

plt.show()
```



```
x = df['year']
y = df['sellingprice']

plt.scatter(x, y, alpha=0.5)
plt.xlabel('Year')
plt.ylabel('Selling Price')
plt.title('Relationship Between Year and Selling Price')

plt.show()
```


Relationship Between Year and Selling Price
Selling Price
Year